2023 American Control Conference (ACC)
San Diego, CA, USA. May 31 - June 2, 2023

Towards Personalized Plasma Medicine via Data-Efficient Adaptation of Fast Deep Learning-based MPC Policies

Kimberly J. Chan†, Georgios Makrygiorgos†, and Ali Mesbah

Abstract— **Plasma medicine has emerged as a promising approach for treatment of biofilm-related and virus infections, assistance in cancer treatment, and treatment of wounds and skin diseases. Despite advances in learning-based and predictive control of plasma medical devices, there remain major challenges towards personalized and point-of-care plasma medicine. In particular, an important challenge arises from the need to adapt control policies after each treatment using (often limited) observations of therapeutic effects that can only be measured between treatments. Control policy adaptation is necessary to account for variable characteristics of plasma and target surfaces across different subjects and treatment scenarios, thus personalizing the plasma treatment to enhance its efficacy. To this end, this paper presents a data-efficient, "globally" optimal strategy to adapt deep learning-based controllers that can be readily embedded on resource-limited hardware for portable medical devices. The proposed strategy employs multi-objective Bayesian optimization to adapt parameters of a deep neural network (DNN)-based control law using observations of closed-loop performance measures. The proposed strategy for adaptive DNN-based control is demonstrated experimentally on a cold atmospheric plasma jet with prototypical applications in plasma medicine.**

I. Introduction

Cold atmospheric plasmas (CAPs) have recently found promising use in plasma medicine [1]. CAPs, a low-temperature (partially) ionized gas, can be generated by applying an electric field to a noble gas, such as argon or helium, whereby the resulting discharge is directed towards a target surface [2]. The synergistic effects of CAPs, including the generation of reactive chemical species and ions, ultraviolet radiation, low-level electric fields, and thermal effects, can induce therapeutic outcomes [3]. As such, portable CAP devices have shown promise for a variety of point-of-care biomedical applications [4]. Nevertheless, CAPs exhibit multivariable, distributed-parameter, and intrinsically variable dynamics and are often subject to (safety-critical) constraints. Thus, there has been a growing interest in advanced control of biomedical CAP devices using model predictive control (MPC) [5] and learning-based control strategies [6]. Two of the main challenges in MPC of CAPs stem from the need to: (i) handle the fast dynamics and, thus, kilohertz (kHz) sampling rates of CAPs [7], and (ii) adapt MPC policy parameters to account for variable characteristics of the plasma and target surfaces [8].

The authors are with the Department of Chemical and Biomolecular Engineering at the University of California, Berkeley, CA 94720, USA. {kchan45,gmakr,mesbah}@berkeley.edu

This work was supported by the US National Science Foundation under Grants 1912772 and 2130734.

†K. J. Chan and G. Makrygiorgos contributed equally to this work.

The notion of learning and adaptation, as well as auto-tuning, of control policies using closed-loop performance data has received increasing attention. Notably, policy-gradient methods [9] have been used as a popular reinforcement learning (RL) approach to guide policy search within continuous control-input spaces, with particular success for MPC policies (e.g., [10]). Due to its use of gradient information, policy-gradient RL is touted as a scalable alternative to the increasingly popular Bayesian optimization (BO) strategy for controller auto-tuning (e.g., [11], [12]), but at a cost of lower data-efficiency, especially when initialized poorly. Instead, BO can be a viable alternative for data-efficient policy search, especially when performance data and/or interactions with the real environment are limited. BO is a derivative-free, probabilistically principled method for "global" optimization that can handle a mixture of continuous, discrete, and categorical decision variables [13]. For example, [14] presents an entropy-search BO approach to use *prior* information from a "cheap" simulated environment for sample-efficient policy learning on the actual physical system. Moreover, the multi-objective nature of policy search can be directly accommodated in BO when there is a need to discover a set of optimal policies due to multiple conflicting objectives [15], [16].

This paper presents a strategy for adaptive deep learning-based approximate MPC, towards personalized and point-of-care biomedical plasma applications. *Approximate* MPC [17], which hinges on approximating MPC laws via offline computations of the optimal control problem, enables control of CAP devices at kHz sampling rates [18]. Deep neural network (DNN)-based approximations of MPC laws are especially attractive due to their low memory footprint and versatile embedded implementations on resource-limited, specialized hardware such as field programmable gate arrays (FPGAs) [19], [20]. For plasma treatment of complex interfaces, it is imperative to adapt control policies to account for the variability among different target surfaces, in addition to the time-varying nature of the plasma and surface characteristics. Moreover, adaptability of the treatment policy is important for personalized plasma medicine, where CAP treatments must be tailored for each individual subject to enhance their therapeutic efficacy without compromising the safety and comfort of patients. However, a key challenge arises from the limited number of treatments/trials that can be performed in a biomedical context, which makes data efficiency a prerequisite for policy adaptation. To this end, we present a multi-objective BO (MOBO) strategy for data-efficient and "globally" optimal adaptation of DNN-

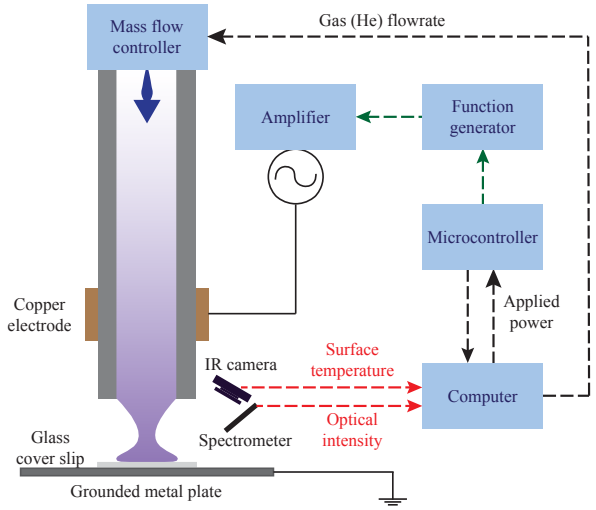

Fig. 1. Schematic of the kHz-excited CAP jet in helium (He). The manipulated inputs are denoted along the black dotted arrows, and the measured outputs are denoted in red.

based control policies in a run-to-run manner. MOBO uses probabilistic surrogate models of multiple closed-loop performance measures (i.e., plasma treatment outcomes) to systematically trade off between exploration and exploitation of a subset of DNN parameters. The selection of this subset of parameters is guided by a global sensitivity analysis that quantifies the influence of each network parameter on the performance measures. As such, MOBO yields a data-efficient scheme for performance-oriented adaptation of DNN-based control policies. We experimentally demonstrate the proposed strategy for adaptive DNN-based approximate MPC of a CAP jet (CAPJ) with prototypical applications in processing of heat-sensitive biomaterials.

II. Robust MPC of Cold Atmospheric Plasma Jet

In this section, we present the control problem for a prototypical CAPJ in the context of personalized plasma treatments. We use a kHz-excited CAPJ in helium (He) that consists of a copper ring electrode wrapped around a quartz tube [21]. A schematic of the CAPJ is shown in Fig. 1. As He gas flows through the tube, plasma ignition is achieved by applying a high-frequency alternating current (AC) voltage to the copper electrode. The plasma is directed out of the tube onto a target substrate, in this case, a grounded, glass-covered metal plate at a distance of 3 mm below the tip of the tube. The applied power P and He flow rate q are the manipulated inputs. The maximum surface temperature T and total optical intensity I of the plasma at the plasma-surface incident point are the measured outputs. Measurements are made available every 0.5 s.

Using data collected from the CAPJ, we model the system dynamics via a linear time-invariant (LTI) state-space model

$$x(k+1) = Ax(k) + Bu(k), \tag{1a}$$
$$y(k) = Cx(k) + Du(k), \tag{1b}$$

where k is the discrete time step, $x \in \mathbb{R}^{n_x}$ is the vector of states, $u = [P, q]^\top \in \mathbb{R}^{n_u}$ is the vector of manipulated inputs, $y = [T, I]^\top \in \mathbb{R}^{n_y}$ is the vector of measured output(s), and A, B, C, D are the state-space matrices identified using subspace identification [22]. The state-space model is defined in terms of deviation variables around a nominal operating condition. Furthermore, we assume an observable, canonical form of (1), where $C = \mathbf{I}$ and $D = \mathbf{0}$. Additionally, we assume that the overall system uncertainty is modeled as a stochastic variable w that is added to (1a).

Plasma treatment of complex surfaces relies on quantification of the delivered plasma effects to a surface. We describe the accumulation of thermal effects on a target with a metric called cumulative equivalent minutes (CEM), as given by

$$\mathrm{CEM}(k+1) = \mathrm{CEM}(k) + K^{(T_{\mathrm{ref}} - T(k))}\delta t, \tag{2}$$

where K is an exponential base dependent on physical properties of the substrate, $T_{\mathrm{ref}} = 43°\mathrm{C}$ is the reference temperature, and δt is the sampling time [23]. This definition of the thermal dose is cumulative in that plasma effects delivered cannot be retracted. With the CEM measure, the augmented system states are $x = [T, I, \mathrm{CEM}]^\top$. Accordingly, the resulting nonlinear model of the CAPJ for thermal treatment of surfaces takes the form

$$x(k+1) = f(x(k), u(k), w(k)). \tag{3}$$

The goal of a plasma treatment is to deliver a desired amount of plasma effects as quickly as possible without violating comfort and safety constraints. Here, we look to systematically account for inherent uncertainties of CAPJs using a robust MPC formulation. To this end, we use scenario-based MPC (sMPC) [24]. sMPC assumes that the system uncertainty is represented by a tree of discrete scenarios, where each branch stemming from a node represents a particular scenario of uncertainty realization. Further, to limit the number of scenarios, a robust horizon N_r is often defined to bound the uncertainty propagation up to a given point [25]. In this work, we select a "worst-case" formulation of the scenario tree, wherein the scenarios are generated based on the worst-case bounds of the uncertainty. To represent the trajectories generated by S scenarios, we adopt the notation $\left(x^j(i), u^j(i)\right)$, where the addition of the superscript j indicates the particular scenario $j \in \{1, \ldots, S\}$. As such, the optimal control problem at time step k is formulated as

$$\min_{x^j, u^j} \quad \sum_{j=1}^{S} p^j V^j(\cdot) \tag{4a}$$
$$\text{s.t.} \quad x^j(i+1) = f\left(x^j(i), u^j(i), w^j(i)\right), \tag{4b}$$
$$\left(x^j(i), u^j(i)\right) \in \mathcal{X} \times \mathcal{U}, \tag{4c}$$
$$x^j(0) = x(k), \tag{4d}$$
$$u^j(i) = u^l(i) \text{ if } x^{b(j)}(i) = x^{b(l)}(i), \tag{4e}$$
$$\forall i \in \{0, \ldots, N_p - 1\},$$

where p^j is the probability of a particular scenario, $V^j(\cdot)$ is the control cost that can consist of a stage cost over a

prediction horizon N_p and/or a terminal cost of a particular scenario; $w^j(i)$ are chosen to be one of three scenarios: zero or proportional to the minimum or maximum error of model identification $[\alpha_1 w_{\min}, 0, \alpha_2 w_{\max}]$, where $\alpha_1, \alpha_2 \in [0,1]$; (4c) are the state and input constraints; and (4e) enforces a *non-anticipativity* constraint, which represents the fact that each control input that branches from the same parent node must be equal ($x^{b(j)}(i)$ is the parent state of $x^j(i+1)$), and l is another counting variable for a particular scenario $l \neq j$. The solution to (4) defines the sMPC law as

$$\pi_{\text{smpc}}\left(x(k)\right) = u^\star(0), \tag{5}$$

where $u^\star(0)$ is the optimal first input. Here, the control objective is defined as the terminal cost

$$V(\text{CEM}(N_p)) = (\text{CEM}_{sp} - \text{CEM}(N_p))^2, \tag{6}$$

where CEM_{sp} denotes the setpoint CEM dose.

Finally, to adapt the policy for personalized CAP treatments, we focus on two closed-loop performance measures: (i) the delivery of a desired amount of thermal dose and (ii) the adherence to a comfort/safety constraint. We define (i) as a CEM tracking cost over the treatment time N

$$\phi_1 = \sum_{k=0}^{N} (\text{CEM}_{sp} - \text{CEM}(k))^2, \tag{7}$$

and (ii) as the sum of the degree of constraint violation in surface temperature over N

$$\phi_2 = \sum_{k=0}^{N} ([T(k) - T_{\text{tol}}]^+)^2, \tag{8}$$

where T_{tol} is the nominal tolerated temperature constraint, and $[T(k) - T_{\text{tol}}]^+$ is the positive magnitude of constraint violation. These measures are competing since T_{tol} is often set to a value at or near $43°$C, which limits the rate of CEM delivery.

III. Approximate MPC using Deep Learning

The requirements of embedded control on low-cost, resource-limited hardware for point-of-use CAPJ applications pose a key challenge to online deployment of the sMPC law (5). The challenge arises from the high computational cost of the scenario-tree optimization in (4). To this end, we use DNNs to approximate (5).

Consider a dataset in the form of

$$\mathcal{T} = \{(x_q, \pi_{\text{smpc}}(x_q))\}_{q=1}^{n_s}, \tag{9}$$

representing n_s state-action (optimal input) pairs as acquired by the offline solution of (4). Let a DNN-based policy be defined as $\Pi : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$. A generic feedforward description of a DNN constitutes a nonlinear, input-output mapping, where information is propagated from the input layer to the output layer via L hidden layers that contain H nodes [26]. Given an input z,

$$\Pi(z; \theta_0, \mathcal{C}) = \boldsymbol{W}_{L+1} \circ (\sigma_L \circ \boldsymbol{W}_L) \circ \cdots \circ (\sigma_1 \circ \boldsymbol{W}_1)(z), \tag{10}$$

where $\theta_0 = \{\boldsymbol{W}_i\}_{i=1}^{L+1}$ are the DNN parameters that are computed via training the DNN using $\mathcal{T}$; $\circ$ denotes the composition operator; and $\mathcal{C}$ denote the hyperparameters. $\boldsymbol{W}_i$ is comprised of the weight matrix and bias between the i-th and $(i+1)$-th layers and σ_i is the activation function.

The DNN parameters are generally fit by minimizing a mean-squared-error loss function. Meanwhile, the DNN hyperparameters related to its architecture (e.g., L, H, $\{\sigma_i\}_{i=1}^{L}$), as well as the training/fitting options (e.g., learning rate, optimizer solver) must be tuned. Tuning is a crucial step of the DNN policy training since the hyperparameters affect the resource utilization of hardware, e.g., the memory required to store the weights/parameters θ_0 and the accuracy of the approximation of (5). While BO is commonly used to facilitate hyperparameter tuning, this work focuses on using BO to adapt DNN parameters. Here, the DNN is trained using closed-loop data as described in, e.g., [19]. This way, each step of the closed-loop trajectory is a solution to (5) and represents a suitable situation in which the closed-loop system is likely to operate.

Remark 1: Adaptation of DNN-based control policies using data-driven optimization methods such as BO is prone to the curse of dimensionality. Thus, we utilize a global sensitivity analysis (GSA) [27] with respect to the performance measures ϕ_1 and ϕ_2 to decide which candidate parameters $\theta \subset \theta_0$ should be prioritized for control policy adaptation.

IV. Multi-objective Bayesian Optimization for Control Policy Adaptation

The control policy adaptation can be cast as a multi-objective (MO) problem characterized by M closed-loop performance measures $\{\phi_m\}_{m=1}^{M}$; see (7), (8). We denote the closed-loop system uncertainties by $\mathbf{d} = \{d(0), \ldots, d(N)\}$. Further, we define a vector-valued performance measure as $\boldsymbol{h}(\theta) : \mathbb{R}^{n_\theta} \to \mathbb{R}^M$, with components $h_m(\theta) = \mathbb{E}_{\mathbf{d}}\left[\phi_m(\theta, \mathbf{d})\right]$, where $\theta \in \Theta$ are real-valued decision variables, namely the subset of DNN parameters that are adapted. Then, the MO optimization problem for optimal selection of θ is formulated as

$$\min_{\theta \in \Theta} \quad \boldsymbol{h}(\theta). \tag{11}$$

We approximate the expectation of each performance measure in a sample-based fashion as

$$h_m(\theta) = \mathbb{E}_{\mathbf{d}}\left[\phi_m(\theta, \mathbf{d})\right] \approx \frac{1}{N_{\text{d}}} \sum_{j=1}^{N_{\text{d}}} \phi_m(\theta, \mathbf{d}_j), \tag{12}$$

where N_{d} is the number of samples for a given θ. The sample-averaged approximation (12) yields noisy estimates of the performance measures

$$\psi_m(\theta) = h_m(\theta) + \epsilon^m, \tag{13}$$

where ϵ^m represents the noise of the m-th performance measure, and $\boldsymbol{\Psi}(\theta) = \{\psi_m(\theta)\}_{m=1}^{M}$ denotes the set of observed performance measures for a given θ.

Problem (11) cannot be directly solved in the case of expensive and black-box performance measures. Hence, the

general idea of BO is to learn probabilistic surrogate models, typically Gaussian process (GP) models, for the performance measures and select a set of points that jointly optimize the expected value of the current surrogates. This is done by solving a proxy problem where an acquisition function proposes points to query in order to refine the surrogate representing the performance measures. The querying strategy is based on the exploration/exploitation trade-off: we look to query the measures at points that lie in a neighborhood that can contain the optima while also reducing the prediction uncertainty of the surrogate models. Given newly observed data $\mathcal{D} = \{(\theta_i, \mathbf{\Psi}_i)\}_{i=1}^{N_o}$, each performance measure is updated using Bayesian inference; e.g., for GP surrogates, GP regression is used [28].

Moreover, in a MO setting, there is not a single best optimizer since the performance measures can be conflicting. Hence, the goal is to discover a set of optimal points, a *Pareto frontier* comprised of *Pareto* optimal points. The Pareto frontier is a boundary in the performance measure space in which improving one measure is realized at the expense of degrading the others. Let us denote a set of Pareto optimal solutions as $\mathcal{P}$. Solutions contained within $\mathcal{P}$ are known to be *non-dominated* by other solutions in the feasible region. For control policy adaptation, *Pareto dominance* is defined as follows.

Definition 1: Given a set of parameters and their corresponding performance measures $\{\theta, \boldsymbol{h}(\theta)\}$, a solution $\boldsymbol{h}(\theta_A)$ dominates another solution $\boldsymbol{h}(\theta_B)$ if $h_i(\theta_A) \leq h_i(\theta_B)$ $\forall i \in \{1, \ldots, M\}$ and $\exists i \in \{1, \ldots, M\}$ such that $h_i(\theta_A) < h_i(\theta_B)$. Pareto dominance is denoted by $\boldsymbol{h}(\theta_A) \prec \boldsymbol{h}(\theta_B)$, while a solution $\boldsymbol{h}(\theta_A)$ is non-dominated if $\nexists \theta_B \in \Theta$ such that $\boldsymbol{h}(\theta_B) \prec \boldsymbol{h}(\theta_A)$.

Given Definition 1, the Pareto frontier is given as

$$\mathcal{P} = \{\boldsymbol{h}(\theta) \text{ s.t. } \nexists \theta^\star \in \Theta : \boldsymbol{h}(\theta^\star) \prec \boldsymbol{h}(\theta)\}, \tag{14}$$

and the set of Pareto optimal parameters is given as

$$\vartheta = \{\theta \in \Theta \text{ s.t. } \boldsymbol{h}(\theta) \in \mathcal{P}\}. \tag{15}$$

Establishing a Pareto frontier will enable the selection of optimal control policies, each of which yields optimal performance with varying levels of trade-offs between the performance measures.

In MOBO, the search for the Pareto frontier is commonly facilitated by the expected *hypervolume* improvement (HVI) acquisition function [29]. The expected HVI relies on the definition of an indicator that quantifies the Pareto optimality of the estimated Pareto frontier known as the hypervolume (HV).

Definition 2: The HV is defined with respect to a reference point $r \in \mathbb{R}^M$ in the performance measure space. For a finite, estimated Pareto set $\mathcal{P}$, the HV is given as the M-dimensional Lebesgue measure Λ_M of the space dominated by $\mathcal{P}$ and bounded by r

$$\mathcal{HV}(\mathcal{P}, r) = \Lambda_M \left(\bigcup_{i=1}^{|\mathcal{P}|} [r, \mathbf{\Psi}_i] \right), \tag{16}$$

TABLE I

SENSITIVITY VALUES (MEAN $\pm$ STANDARD ERROR) OF THE CLOSED-LOOP MEASURES TO VARIOUS PARAMETERS OF THE POLICY

	First Layer	Last Layer
Thermal Dose Delivery (ϕ_1)	$0.025 \pm 3.2\text{e}{-4}$	$0.026 \pm 8.3\text{e}{-4}$
Temperature Constraint (ϕ_2)	$0.036 \pm 4.7\text{e}{-4}$	$0.038 \pm 1.1\text{e}{-3}$

where $|\mathcal{P}|$ is the cardinality of $\mathcal{P}$, and $[r, \mathbf{\Psi}_i]$ is the hyperrectangle formed by the points r and $\mathbf{\Psi}_i$ [29].

The HV acts as a metric to quantify the quality of the Pareto frontier and is affected by the selection of the reference point. Thus, "convergence" to a single HV value means that MOBO has performed enough sampling (based on some prespecified budget) to construct the best possible Pareto frontier, which is not necessarily the true one. Then, the HVI describes the incremental improvement of the HV of $\mathcal{P}$ if a new point is added. The HVI of a set of newly observed measures $\mathbf{\Psi}'$ is given by

$$\mathcal{HVI}(\mathbf{\Psi}', \mathcal{P}, r) = \mathcal{HV}(\mathbf{\Psi}' \cup \mathcal{P}, r) - \mathcal{HV}(\mathcal{P}, r). \tag{17}$$

Hence, the expected HVI acquisition function α_{EHVI} describes the expectation of HVI over the posterior of the performance measures and is given as

$$\alpha_{\text{EHVI}}(\theta) = \mathbb{E}[\mathcal{HVI}(\mathbf{\Psi}', \mathcal{P}, r)]. \tag{18}$$

Finally, to account for noisy observations of performance measurements, the noisy expected HVI acquisition (NEHVI) is employed; see [29]. Here, NEHVI is maximized with respect to the DNN parameters θ.

V. ADAPTIVE DNN-BASED CONTROL POLICIES FOR PERSONALIZED PLASMA TREATMENTS

We demonstrate the proposed MOBO strategy for control policy adaptation on the CAPJ described in Section II.

A. Control Policy Approximation

First, we solved the sMPC problem (4) in closed loop to gather training data for approximating the initial control law (5). In (4), we set the prediction horizon $N_p = 5$, the robust horizon $N_r = 2$, and the discrete uncertainty scenarios as $[0.01w_{\min}, 0, 0.01w_{\max}]$. The control inputs are constrained by $P \in [1.5, 5]$ W and $q \in [1.5, 5]$ SLM, and the states are constrained by $T \in [25, 45]^\circ$C and $I \in [20, 80]$ arb. units. The sMPC was formulated using CasADi [30] and solved with IPOPT [31]. We simulated the true system with a mismatch between the plant and control model and normally distributed measurement noise $\mathcal{N}(0, (0.1)^2)$. We collected $n_s = 5,000$ samples of state-to-optimal-input mappings and trained a fully-connected feedforward DNN architecture with $L = 4$, $H = 7$, and ReLU activation functions. We trained the DNN for $5,000$ epochs using PyTorch [32] with the default optimizer settings. The resulting DNN-based policy achieved nearly equivalent performance to the implicit sMPC law. Furthermore, the computation time of the DNN, which depends on the architecture of the DNN, compared

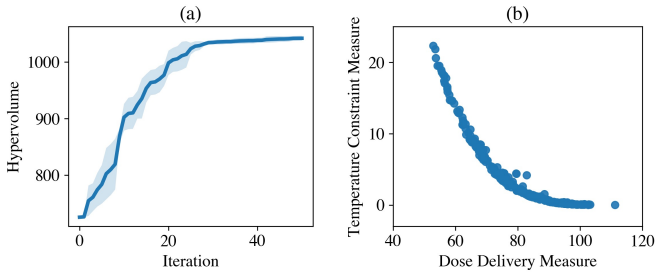

Fig. 2. (a) Hypervolume improvement (mean ± two standard errors) and (b) observed Pareto frontier over five replicate runs of MOBO. The hypervolume improvement (a) demonstrates that MOBO reaches some optimal representation of the Pareto frontier. The Pareto frontier (b) demonstrates the trade-off between the competing performance measures (dose delivery: reducing treatment time; temperature constraint: satisfying patient comfort and safety).

to solving (4), on a standard CPU (2.4 GHz quad-core Intel i5 processor) was roughly three orders of magnitude faster ($\sim 10^{-5}$ s versus $\sim 10^{-2}$ s).

B. Control Policy Adaptation in Closed-loop Simulations

We consider the treatment of a subject with characteristics that differ from the mean population values, and our goal is to adapt the initial policy designed for the population mean to cater to the individual subject. Note that the closed-loop performance measures are parameterized by subject-specific characteristics, namely K in the CEM setpoint tracking cost (7) and T_{tol} in the comfort constraint cost (8). Here, we examine the case in which the parameters of the population mean are $K_{\text{pop}} = 0.5$ and $T_{\text{tol,pop}} = 45°\text{C}$, while the characteristics of the individual subject are $K_{\text{indiv}} = 0.55$ and $T_{\text{tol,indiv}} = 44.5°\text{C}$.

1) Selection of Adaptation Parameters: First, we examine the sensitivity of the DNN-based policy to perturbations in different subsets of its parameters. Knowing that the desire for personalized treatments is to minimize the number of trial-and-error treatments, we adapted a subset of DNN parameters due to its high dimensionality ($n_{\theta_0} = 212$). Common practice is to freeze the DNN and adapt the last layer and/or append a new layer to the network to adapt. To evaluate this practice, we examine the sensitivity of the closed-loop performance measures to the parameters of the first and last layers of the DNN-based policy. To perform a GSA as described in Remark 1, we used the sensitivity analysis tools by UQLab [33]. We used a moment-independent method (i.e., Borgonovo indices) to analyze the global sensitivity of the selected DNN parameters to the closed-loop performance measures. We generated $10,000$ samples of the 44 parameters encapsulated by the first and last layers of the 4-layer, 7-node DNN. Samples were selected from geometrically-bounded values from the initial policy parameters. For each sample, we ran triplicate closed-loop simulations using the DNN-based policy and evaluated the mean plus and minus standard error values of the observed closed-loop measures. Table I lists the results of the GSA. In general, the dose delivery measure (7) is less sensitive to changes in the parameters compared to the temperature constraint measure (8). Overall, both of the measures are equally sensitive to all of the parameters selected for GSA. Despite this, the parameters of the last layer have slightly higher influence with fewer number of parameters. Hence, we selected the last layer of the DNN-based policy as the subset of parameters to modify in our policy search procedure (i.e., $\theta = \boldsymbol{W}_{L+1}$ such that $n_\theta = 16$).

2) Personalized Control Policies: As a global optimization method, MOBO provides a means to systematically explore and detect trade-offs between competing performance measures. Fig. 2 shows the results of 5 replicates of 50 iterations of MOBO on a simulated CAPJ, where one iteration of MOBO is comprised of $N_{\mathbf{d}} = 3$ replicates.[1] The HV profile in Fig. 2(a) shows the "convergence" of MOBO. The reduced improvement in the HV after more than 30 iterations indicates that MOBO has achieved some optimal representation of the Pareto frontier depicted in Fig. 2(b). While it took 20 or more iterations to achieve this Pareto frontier, the first few iterations of MOBO can drastically improve upon the initial policy. The steep increase in HV suggests that the initial policy parameterization is suboptimal, and a new Pareto optimal point can be found in the first few iterations even when starting with a suboptimal solution.

C. Control Policy Improvement for Real-time Treatments

For the experimental demonstrations of the proposed approach on the CAPJ depicted in Fig. 1, the sMPC had $N_p = N_r = 2$, and the input bounds were adjusted to $P \in [1.5, 3.5]$ W and $q \in [3.5, 7.5]$ SLM. Then, 11 closed-loop experiments resulting in $n_s = 1,378$ samples were performed to gather training data for the DNN approximation. The DNN was trained with the same structure and procedure as described for the simulation studies and achieved similar closed-loop performance to implicit sMPC. MOBO was performed for 15 total iterations due to a limited budget of 45 treatments.

Fig. 3 shows the state and input profiles of 3 particular iterations of MOBO. Within each iteration, we performed $N_{\mathbf{d}} = 3$ replicate real-time experiments to account for the intrinsic variability of the system. The CEM profiles are plotted with min-max bounds of the three replicates represented by the shaded region, while the solid line represents the median value of the triplicate runs. The temperature profiles represent the mean value (solid lines) plus and minus two standard errors (shaded region) of the triplicate experiments. Both input profiles are plotted with the mean value from the triplicate experiments. In Fig. 3, the profiles shown are determined to be a few of the "best" treatment options encountered through the process of MOBO. In this case, the "best" is described in one of two ways: (i) if there was insufficient data to establish a clear Pareto frontier, then

[1]To implement MOBO, we used Ax [34]. Ax interfaces with BoTorch [35] to perform BO, and BoTorch interfaces with GPyTorch [36] for the surrogate modeling with GPs. These tools were primarily used with their default settings, using the Matern 5/2 kernel for GPs and the noisy EHVI acquisition function. Codes are available at `https://github.com/kchan45/BO4Policy_Search_Plasma`.

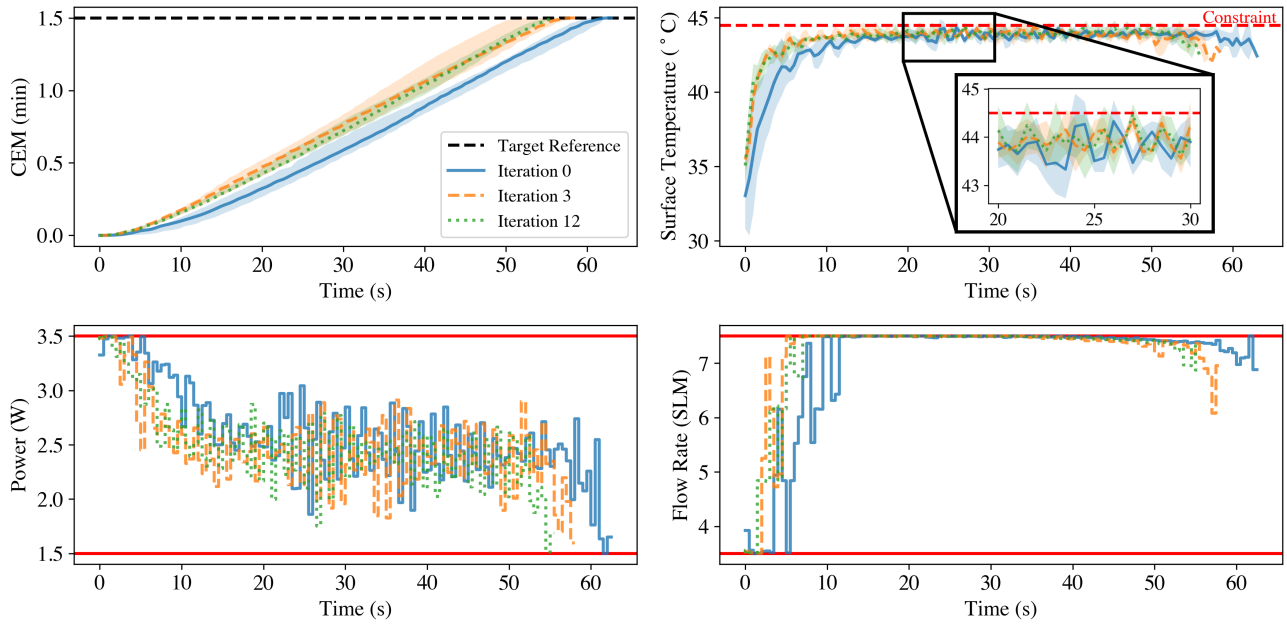

Fig. 3. State and input profiles of closed-loop experiments at various iterations of MOBO. Each iteration of MOBO consisted of triplicate experiments. The CEM profile (upper left) shows the median value (solid line) along with the min/max range (shaded region). The surface temperature profile (upper right) shows the mean value (solid line) and two standard errors (shaded region). For the manipulated inputs (power and flow rate), only the mean value is plotted. The selected profiles shown are designated as the trajectories that correspond to the "incumbent best" policy parameterizations. The incumbent best is deemed as the initial policy, if a Pareto frontier cannot be established (i.e., in the first few iterations) or the policy parameterization on the Pareto frontier with the lowest temperature constraint measure.

the best treatment was considered the initial policy, and (ii) once an estimated Pareto optimal point was found, the best treatment used the policy that produces the lowest constraint violation. This sequence of treatment protocols follows a "safe" treatment intuition. As in for (i), the initial treatment is deemed safe for the general population and is considered "best" for the time being. In the case of (ii), once a Pareto frontier is established, the treatment may then be switched to a more optimal one at the cost of minor temperature violations. Note that establishing some trade-off between the different performance measures via an estimated Pareto frontier allows for the personalization of plasma treatments.

From Fig. 3, the first "best" profile is the initial profile (in blue); a new "best" is encountered after Iteration 3 (in dashed orange). The dashed orange profile represents a new parameterization of the policy that outperforms the initial blue policy, as it achieves the CEM faster (reducing the median treatment time by roughly 8 s or 13%), with slight constraint violations. After more iterations of MOBO, a new policy in dotted green is found at Iteration 12. In this case, the CEM delivery on average is similar to the orange policy (reducing the median treatment time from the initial policy by 10 s or 16%), while maintaining a lower constraint cost. Furthermore, in Fig. 3, the flow rate of He tends to become saturated during treatment. In general, higher He flow rates are characteristic of lower temperatures. Thus, because of the temperature tolerance specification, the operation of the CAPJ necessitates higher flow rate to remain within the region of desirable operating temperatures. The locations

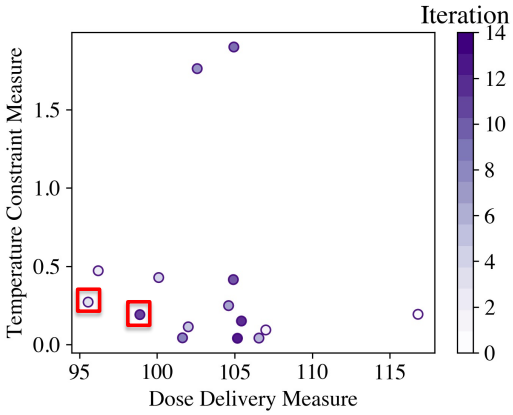

Fig. 4. Observed performance measures from the MOBO exploration. A total of 15 iterations of MOBO were performed. Each individual point represents the mean performance measure values from triplicate real-time experiments at each iteration. The red boxes identify the estimated Pareto optimal points.

of these "best" points in the performance measure space is shown by the red boxes in Fig. 4. With few iterations, the Pareto frontier cannot be visually established, but several points may still be identified as Pareto optimal by their proximity to the minimization of both performance measures. Fig. 4 is consistent with the profiles in Fig. 3 in that Pareto optimal points are found at Iterations 3 and 12 (as indicated by the red boxes). As such, a strategy has been established that can trade off between multiple performance measures in order to tailor the treatment to individual subjects.

VI. Conclusions and Future Work

This paper presented multi-objective BO (MOBO) as an effective strategy for the adaptation of deep learning-based controllers towards personalized plasma medicine. We discussed how a computationally expensive (robust) MPC law can be approximated with deep learning, where global sensitivity analysis is utilized to limit the number of parameters to be adapted. We demonstrated the capability of MOBO in efficiently exploring the parameter space of a neural network controller in closed-loop simulations and in real-time experiments. Our future work will involve embedding approximate MPC policies on resource-limited hardware, towards adaptable MPC-on-a-chip for portable plasma devices.

References

[1] M. Laroussi, "Plasma medicine: A brief introduction," *Plasma*, vol. 1, no. 1, pp. 47–60, 2018.

[2] X. Lu, M. Laroussi, and V. Puech, "On atmospheric-pressure non-equilibrium plasma jets and plasma bullets," *Plasma Sources Science and Technology*, vol. 21, no. 3, p. 034005, 2012.

[3] D. Petlin, S. Tverdokhlebov, and Y. Anissimov, "Plasma treatment as an efficient tool for controlled drug release from polymeric materials: A review," *Journal of Controlled Release*, vol. 266, pp. 57–74, 2017.

[4] M. Laroussi, S. Bekeschus, M. Keidar, A. Bogaerts, A. Fridman, X. Lu, K. Ostrikov, M. Hori, K. Stapelmann, V. Miller, *et al.*, "Low-temperature plasma for biology, hygiene, and medicine: Perspective and roadmap," *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 6, no. 2, pp. 127–157, 2022.

[5] Y. Lyu, L. Lin, E. Gjika, T. Lee, and M. Keidar, "Mathematical modeling and control for cancer treatment with cold atmospheric plasma jet," *Journal of Physics D: Applied Physics*, vol. 52, no. 18, p. 185202, 2019.

[6] L. Lin and M. Keidar, "Machine learning controlled self-adaptive plasma medicine," in *Proceedings of the IEEE International Conference on Plasma Science*, pp. 561–561, 2020.

[7] A. D. Bonzanini, J. A. Paulson, G. Makrygiorgos, and A. Mesbah, "Fast approximate learning-based multistage nonlinear model predictive control using Gaussian processes and deep neural networks," *Computers & Chemical Engineering*, vol. 145, p. 107174, 2021.

[8] A. D. Bonzanini, K. Shao, A. Stancampiano, D. B. Graves, and A. Mesbah, "Perspectives on machine learning-assisted plasma medicine: Toward automated plasma treatment," *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 6, no. 1, pp. 16–32, 2021.

[9] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the International Conference on Machine Learning*, pp. 387–395, Pmlr, 2014.

[10] M. Zanon and S. Gros, "Safe reinforcement learning using robust MPC," *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3638–3652, 2020.

[11] D. Piga, M. Forgione, S. Formentin, and A. Bemporad, "Performance-oriented model learning for data-driven MPC design," *IEEE Control Systems Letters*, vol. 3, no. 3, pp. 577–582, 2019.

[12] F. Sorourifar, G. Makrygirgos, A. Mesbah, and J. A. Paulson, "A data-driven automatic tuning method for MPC under uncertainty using constrained Bayesian optimization," *IFAC-PapersOnLine*, vol. 54, no. 3, pp. 243–250, 2021.

[13] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[14] A. Marco, F. Berkenkamp, P. Hennig, A. P. Schoellig, A. Krause, S. Schaal, and S. Trimpe, "Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1557–1563, 2017.

[15] G. Makrygiorgos, A. D. Bonzanini, V. Miller, and A. Mesbah, "Performance-oriented model learning for control via multi-objective Bayesian optimization," *Computers & Chemical Engineering*, vol. 162, p. 107770, 2022.

[16] M. Turchetta, A. Krause, and S. Trimpe, "Robust model-free reinforcement learning with multi-objective Bayesian optimization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 10702–10708, 2020.

[17] A. Mesbah, K. P. Wabersich, A. P. Schoellig, M. N. Zeilinger, S. Lucia, T. A. Badgwell, and J. A. Paulson, "Fusion of machine learning and MPC under uncertainty: What advances are on the horizon?," in *Proceedings of the American Control Conference*, pp. 342–357, 2022.

[18] A. D. Bonzanini, J. A. Paulson, D. B. Graves, and A. Mesbah, "Toward safe dose delivery in plasma medicine using projected neural network-based fast approximate NMPC," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5279–5285, 2020.

[19] K. J. Chan, J. A. Paulson, and A. Mesbah, "Deep learning-based approximate nonlinear model predictive control with offset-free tracking for embedded applications," in *Proceedings of the American Control Conference*, pp. 3475–3481, 2021.

[20] S. Lucia, D. Navarro, B. Karg, H. Sarnago, and O. Lucia, "Deep learning-based model predictive control for resonant power converters," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 409–420, 2020.

[21] D. Gidon, D. B. Graves, and A. Mesbah, "Effective dose delivery in atmospheric pressure plasma jets for plasma medicine: A model predictive control approach," *Plasma Sources Science and Technology*, vol. 26, no. 8, p. 085005, 2017.

[22] P. Van Overschee and B. De Moor, *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media, 2012.

[23] S. A. Sapareto and W. C. Dewey, "Thermal dose determination in cancer therapy," *International Journal of Radiation Oncology Biology Physics*, vol. 10, no. 6, pp. 787–800, 1984.

[24] D. Bernardini and A. Bemporad, "Scenario-based model predictive control of stochastic constrained linear systems," in *Proceedings of the 48th IEEE Conference on Decision and Control*, pp. 6333–6338, 2009.

[25] S. Lucia, T. Finkler, and S. Engell, "Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty," *Journal of Process Control*, vol. 23, no. 9, pp. 1306–1319, 2013.

[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[27] E. Borgonovo and E. Plischke, "Sensitivity analysis: A review of recent advances," *European Journal of Operational Research*, vol. 248, no. 3, pp. 869–887, 2016.

[28] C. E. Rasmussen, C. K. Williams, *et al.*, *Gaussian processes for machine learning*, vol. 1. Springer, 2006.

[29] S. Daulton, M. Balandat, and E. Bakshy, "Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement," *Advances in Neural Information Processing Systems*, vol. 34, pp. 2187–2200, 2021.

[30] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.

[31] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 2006.

[32] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "PyTorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[33] S. Marelli and B. Sudret, "UQLab: A framework for uncertainty quantification in Matlab," in *Vulnerability, Uncertainty, and Risk: Quantification, Mitigation, and Management*, pp. 2554–2563, 2014.

[34] E. Bakshy, L. Dworkin, B. Karrer, K. Kashin, B. Letham, A. Murthy, and S. Singh, "AE: A domain-agnostic platform for adaptive experimentation," in *Proceedings of the Conference on Neural Information Processing Systems*, 2018.

[35] M. Balandat, B. Karrer, D. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy, "BoTorch: A framework for efficient Monte-Carlo Bayesian optimization," *Advances in Neural Information Processing Systems*, vol. 31, pp. 21524–21538, 2020.

[36] J. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson, "GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration," *Advances in Neural Information Processing Systems*, vol. 31, 2018.